UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/650,363	08/27/2003	Harshvardhan Sharangpani	81862P288	7318

8791 7590 12/27/2007

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
TIMBLIN, ROBERT M	
ART UNIT	**PAPER NUMBER**
2167	

MAIL DATE	DELIVERY MODE
12/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/650,363	SHARANGPANI ET AL.	
	Examiner	Art Unit	
	Robert M. Timblin	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 October 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25 and 27-72* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25 and 27-72* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☒ Interview Summary (PTO-413) Paper No(s)/Mail Date. *20071219* .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This Office Action corresponds to application 10/650, filed 8/27/2003.

Applicant Requested Interview

On December 19, 2007, an interview was held regarding the present application. The participants in the interview included Examiner Robert Timblin and Applicant's representative, Neal Berezny. Substance of the interview is attached herein.

Response to Amendment

Claims 1, 67, and 71 have been amended. Accordingly, claims 1-25 and 27-72 are pending prosecution.

Claim Objections

Claims 48-66 are objected to because of the following informalities: these claims, which depend at least in part on claim 47 recite a "rule engine content processor". As per the amendment submitted 8/31/2006, claim 47 was amended to remove "engine content" from [the] "ruled engine content processor". Accordingly, "engine content" should be removed from claims 48-66.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country; more than one year prior to the date of application for patent in the United States.

Claims 1-3, 6, 9-23, 25, 27-32, 35-37, 39-48, 51-60, and 62-72 are rejected under 35 U.S.C. 102(b) as being anticipated by Lee U.S. Patent 5,060,143. In the following citations and figures, Lee teaches and describes:

With respect to claim 1, A rule processor for conducting contextual searches, the processor comprising:

a plurality of M input payload search registers (figure 1, drawing reference 102, and s[j]), wherein a data stream of content data (col. 2 line 18-19) to be searched (i.e. a candidate database stream) is input (i.e. loaded) into the plurality of payload search registers (figure 1, drawing reference 102, and s[j]);

a search execution engine (drawing reference 100) comprising:

a search array (figure 1, comparator array 140) coupled to the plurality of M search registers (figure 1, drawing reference 102, and s[j]), wherein the search array comprises:

a plurality of M rows (figure 1, P[i]) of search array elements (figure 1, points of computation) coupled to a plurality of M output match lines (figure 1, drawing references 10, 12, and col. 2 line 33-37, match trace; i.e. Lee discloses a match line for every point of computation); and

a plurality of N columns (s[i], c[i]) of search array elements (figure 1, points of computation) coupled to a plurality of N pattern input lines (col. 5 line 57, col. 7 line 34-37) comprising a search pattern (col. 4 line 4; e.g. a target pattern), wherein the search

array (figure 1, comparator array 140) comprises an array of M by N search array elements (figure 1, see the comparator array in each stage 1-3 as a two dimensional (i.e. M by N) array), and wherein the content data (col. 2 line 18-19) in the plurality of M search registers (figure 1, drawing reference 102, and s[j]) is replicated and stored N times in the plurality of N columns in the search array (col. 2 line 18-22); and

a sorter (figure 3, drawing reference 120) coupled to the search array (figure 1, comparator array 140) to perform one or more contextual searches (col. 3 line 37-40) on content in the search array (140) via parallel pattern matching (col. 2 line 26-27 and col. 4 line 15-17) in response to executing one or more search instructions (figure 2, instructions) specifying the one or more pattern searches (col. 7 line 42-45) and presenting one or more patterns to the content (figure 1; i.e. presenting "filters" to the input candidate stream), wherein the parallel pattern matching (col. 2 line 26-27 and col. 4 line 15-17) comprises performing a simultaneous search (col. 7 line 1-6) within all M rows (figure 1, p[i]) for the search pattern (col. 4 line 4; e.g. a target pattern and "filters") input by the N pattern input lines (col. 5 line 57, col. 7 line 34-37) in one clock period (figure 1, stages 1, 2, or 3, col. 2 line 13-29, and col. 4 line 12-19; e.g. Lee discloses comparing all characters in "file" are compared with the target pattern in a single stage (i.e. a clock cycle).

With respect to claim 2, The rule processor defined in Claim 1 wherein fields of the one or more search instructions (figure 2, instructions) are coupled to the plurality of

search registers (figure 1, drawing reference 102, and s[j]) and the search execution engine (drawing reference 100).

With respect to claim 3, The rule processor defined in Claim 1 wherein at least one of the one or more search instructions specifies a pattern that is to be searched against the content in the search array and zero or more search parameters (col. 7 line 43).

With respect to claim 6, The rule processor defined in Claim 4 wherein the zero or more parameters specify starting and ending locations that constitute a range of the content within the search array within which the search execution engine is to constrain a search (col. 7 line 30-35).

With respect to claim 9, The rule processor defined in Claim 1 wherein the search execution engine generates at least one result output indicative of success in searching the content in the search array (co. 3 line 21-23; i.e. a "HIT").

With respect to claim 10, The rule processor defined in Claim 4 wherein the at least one result output comprises an indication of whether or not a match occurred between a pattern specified in at least one of the instructions and the content in the search array (stage 3, figure 1).

With respect to claim 11, The rule processor defined in Claim 4 wherein the at least one result output comprises an indication of a location in the search array where a match occurred between a pattern specified in at least one of the instructions and the content in the search array (col. 6 line 3, vector 'V' containing position data).

With respect to claim 12, The rule processor defined in Claim 1 wherein at least one search instruction includes a field that specifies a parameter to use to control the search or a pointer into a memory that stores the parameter to control the search (col. 7 line 43-45).

With respect to claim 13, The rule processor defined in Claim 12 wherein the pointer points to a general purpose register (drawing reference 110).

With respect to claim 14, The rule processor defined in Claim 12 wherein a value to which the pointer points is a result of a previously performed search by the search execution hardware (col. 2 line 24-25).

With respect to claim 15, The rule processor defined in Claim 12 wherein the parameter corresponds to one of a group that includes a mask, a search window parameter, and a control parameter (col. 7 line 56; i.e. control signals).

With respect to claim 16, The rule processor defined in Claim 1 wherein the plurality of input payload search registers comprises a register file (drawing reference 109).

With respect to claim 17, The rule processor defined in Claim 16 wherein the register file comprises 2K entries of one byte each (col. 5 line 41-45).

With respect to claim 18, The rule processor defined in Claim 16 wherein the register file comprises a plurality of entries addressed by 11-bit register addresses (figure 2).

With respect to claim 19, The rule processor defined in Claim 1 further comprising a memory to store one or more search instructions to be applied to content in the search array (drawing reference 110).

With respect to claim 20, The rule processor defined in Claim 1 wherein the search instructions cause the search execution engine to perform searches for arbitrarily long patterns in the content in the search array (col. 7 line 43; i.e. a user specified pattern to search).

With respect to claim 21, The rule processor defined in Claim 1 further comprising an instruction sequencer for applying one or more search instructions to the search execution engine (drawing reference 110, instruction engine unit).

With respect to claim 22, The rule processor defined in Claim 21 wherein the one or more search instructions specify at least one pattern (col. 7 line 43), range control (p[1-7], and program control flow (col. 7 line 56).

With respect to claim 23, The rule processor defined in Claim 21 wherein the one or more search instructions include a pointer to specify a memory location that stores information (110) that specifies at least one pattern (col. 7 line 43), range control (p[1-7], and program control flow (col. 7 line 56).

With respect to claim 25, The rule processor defined in Claim 1 wherein the search execution engine comprises a first output indication indicative of search success of execution of one search instruction (col. 3 line 21-23) and a second output indication indicative of a location within the search registers of a pattern specified by the one search instruction (col. 6 line 1-4).

With respect to claim 27, The rule processor defined in Claim 1 wherein the search array comprises M match lines with each of the M match lines (figure 1, drawing references 10, 11, and col. 2 line 33-37, match trace; i.e. Lee discloses a match line for

every point of computation) associated with a group of data stored in the search array (figure 1, comparator array 140) and being indicative of whether a pattern specified by one of the one or more search instructions matches data in its associated group of data stored in the search array (figure 1, e.g. a "HIT" indication).

With respect to claim 28, The rule processor defined in Claim 27 wherein the sorter is coupled to receive the M match lines to perform the one or more operations associated with matches indicated by the M match lines (figure 1; e.g. processing "early out" or "HIT" indicators).

With respect to claim 29, The rule processor defined in Claim 28 wherein the information specifies a range (S[1-12], and the sorter sorts the M match lines only in the specified range (figure 1).

With respect to claim 30, The rule processor defined in Claim 29 wherein the range is specified in the search instruction (P[1-7]).

With respect to claim 31, The rule processor defined in Claim 29 wherein the information specifies a location in a memory at which the range is stored (col. 7 line 29; i.e. address information).

With respect to claim 32, The rule processor defined in Claim 31 wherein the memory is a register file (drawing reference 106).

With respect to claim 35, The rule processor defined in Claim 1 wherein the sorter has a first output indicating whether one or more of the match lines match (e.g. "HIT") and a second output indicative of a result of performing the one or more operations (col. 6 line 1-4).

With respect to claim 36, The rule processor defined in Claim 35 wherein the second output is indicative of a location into the search array of a first occurrence of a match between the pattern and data stored in the search array in relation to one side of the search array (col. 6 line 1-4; i.e. position information).

With respect to claim 37, The rule processor defined in Claim 35 wherein the second output is indicative of a number of matches in a range of the M match lines (fig. 1 e.g. finding 3 matches to "fil").

With respect to claim 39, The rule processor defined in Claim 1 wherein the sorter further comprises:

a priority encoder to identify a location in the search array corresponding to the M match lines (figure 1, drawing references 10, 11, and col. 2 line 33-37, match trace; i.e. Lee discloses a match line for every point of computation) corresponding to a first

occurrence of a match between the pattern and data stored in the search array in relation to one side of the search array (col. 6 line 1-5).

With respect to claim 40, The rule processor defined in Claim 39 wherein the priority encoder is an ascending priority encoder and the one side of the search array is the top of the search array (figure 1, stage 1 and P[1] as the top).

With respect to claim 41, The rule processor defined in Claim 39 wherein the priority encoder is a descending priority encoder and the one side of the search array is the bottom of the search array (figure 1, stages 2-3 and P[4] or p[7] as the bottom).

With respect to claim 42, The rule processor defined in Claim 1 wherein the sorter further comprises a counter to determine a number of matches in the search array (col. 2 line 54; e.g. finding zero results).

With respect to claim 43, The rule processor defined in Claim 1 wherein the sorter further comprises:

an ascending priority encoder to identify a location in the search array corresponding to the M match lines (figure 1, drawing references 10, 11, and col. 2 line 33-37) corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a top side (figure 1, P[1]) of the search array (figure 1, stage 1);

a descending priority encoder to identify a location in the search array corresponding to the M match lines (figure 1, drawing references 10, 11, and col. 2 line 33-37) corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a bottom side (stage 1, P[3] or stage 3 P[7]) of the search array (figure 1, stages 2-3);

a counter to determine a number of matches in a range of the M match lines (col. 2 line 54; e.g. finding zero results); and

a selector coupled to the ascending priority encoder, the descending priority encoder and the counter and having a first output, the selector operable to select an output of the ascending priority encoder (figure 1, stage 1), the descending priority encoder (figure 1, stages 2-3) and the counter as the first output of the sorter (figure 1 and vector 'V').

With respect to claim 44, The rule processor defined in Claim 43 wherein the selector has a second output indicating if a match occurred between the pattern and data in the search array (figure 1; e.g. "HIT" indication).

With respect to claim 45, The rule processor defined in Claim 1 wherein the search array comprises:

a plurality of rows of memory locations to store bytes of data (figure 1, P[j];

a plurality of rows of byte comparators (figure 1, c1-c4) to compare bytes of the data stored in the plurality of rows of memory locations with bytes of the pattern (e.g.

"filters"), each comparator (figure 1, c1-c4) of the plurality of rows of byte comparators having an output (10, 12);

a plurality of masked reduction units, each of the plurality of masked reduction units coupled to receive byte masks and comparator outputs (e.g. a "HIT" or EROF") of comparators (c1-c4) in one row of byte comparators (e.g. P[1]), the plurality of masked reduction units masking individual comparator outputs based on the byte masks and combining unmasked comparator outputs for each row into one of a plurality of mask lines (figure 1, stage 1; e.g. Lee shows if an early out flag were encountered the remaining pattern (i.e. P[5-7] would not be search and in effect, masked).

With respect to claim 46, The rule processor defined in Claim 1 further comprising:

a rule memory to store a plurality of rules (drawing reference 110);

a rule sequencer coupled to the rule memory to select one or more rules for execution (col. 7 line 49-53, figure 1); and

a decoder (110) to decode the one or more rules selected by the rule sequencer, the decoder coupled to the search array and sorter to provide decoded information to the search array and the sorter (col. 7 line 43).

With respect to claim 47, The rule processor of claim 1, wherein the search array performs pattern matching between data stored in the search array (140) and an N byte pattern from a search instruction (figure 2, instructions) received on a first input (124), the search array (140) having M match lines (figure 1, drawing references 10, 12, and col. 2

line 33-37, match trace; i.e. Lee discloses a match line for every point of computation) as outputs with each of the M match lines (figure 1, drawing references 10, 12, and col. 2 line 33-37, match trace; i.e. Lee discloses a match line for every point of computation) associated with a group of data stored in the array (e.g. candidate data stream) and being indicative of whether the N byte pattern matches data stored in its associated group of data stored in the search array (e.g. "HIT"), and wherein the sorter receives the M match lines (figure 1, drawing references 10, 12, and col. 2 line 33-37) to perform one or more operations associated with matches indicated by the M match lines (figure 1, drawing references 10, 12, and col. 2 line 33-37), the one or more operations being performed in response to information specified by the rule, and further wherein the sorter outputs data indicative of any match found (e.g. "HIT").

With respect to claim 48, The rule engine content processor defined in claim 47, wherein the sorter has a first output indicating whether one or more of the match lines match and a second output indicative of a result of performing the one or more operations (figure 1, "HIT").

With respect to claim 51, The rule engine content processor defined in Claim 50 wherein the range mask logically ANDs the M mask lines with a pair of offsets specified by the search instruction (col. 6 line 17-18).

With respect to claim 52, The rule engine content processor defined in Claim 51 wherein the rule includes the pair of offsets (col. 7 line 30; i.e. address information).

With respect to claim 53, The rule engine content processor defined in Claim 50 wherein the rule includes a pointer to a location in a memory where the offsets are stored (106).

With respect to claim 54, The rule engine content processor defined in claim 47 wherein the data output for the sorter is feedback for use and an input to the sorter in the next cycle (figure 1, stage 2-3 and use of the 'V' register).

With respect to claim 55, The rule engine content processor defined in Claim 47 wherein the sorter further comprises:

a priority encoder to identify a location in the search array corresponding to the M match lines (figure 1, drawing references 10, 11, and col. 2 line 33-37, match trace; i.e. Lee discloses a match line for every point of computation) corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to one side of the search array (col. 6 line 1-5).

With respect to claim 56, The rule engine content processor defined in Claim 55 wherein the priority encoder is an ascending priority encoder and the one side of the search array is the top of the search array (figure 1, stage 1 and P[1] as the top).

With respect to claim 57, The rule engine content processor defined in Claim 55 wherein the priority encoder is a descending priority encoder and the one side of the search array is the bottom of the search array (figure 1, stages 2-3 and P[4] or p[7] as the bottom).

With respect to claim 58, The rule engine content processor defined in Claim 47 wherein the sorter further comprises a counter to determine a number of matches in a range of the M match lines (col. 2 line 54; e.g. finding zero results).

With respect to claim 59, The rule engine content processor defined in Claim 47 wherein the sorter further comprises:

an ascending priority encoder to identify a location in the search array corresponding to the M match lines (figure 1, drawing references 10, 11, and col. 2 line 33-37) corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a top side (figure 1, P[1]) of the search array (figure 1, stage 1);

a descending priority encoder to identify a location in the search array corresponding to the M match lines (figure 1, drawing references 10, 11, and col. 2 line 33-37) corresponding to a first occurrence of a match between the pattern and data stored in the search array in relation to a bottom side (stage 1, P[3] or stage 3 P[7]) of the search array (figure 1, stages 2-3);

a counter to determine a number of matches in a range of the M match lines (col. 2 line 54; e.g. finding zero results); and

a selector coupled to the ascending priority encoder, the descending priority encoder and the counter and having a first output, the selector operable to select an output of the ascending priority encoder (figure 1, stage 1), the descending priority encoder (figure 1, stages 2-3) and the counter as the first output of the sorter (figure 1 and vector 'V').

With respect to claim 60, The rule engine content processor defined in Claim 55 wherein the selector has a second output indicating if a match occurred between the pattern and data in the search array (figure 1; e.g. "HIT" indication).

With respect to claim 62, The rule engine content processor defined in Claim 47 wherein the information specifies a range (e.g. P[1-7]), and the sorter sorts the M match lines only in the specified range (figure 1).

With respect to claim 63, The rule engine content processor defined in Claim 62 wherein the range is specified in the search instruction (figure 2, instructions).

With respect to claim 64, The rule engine content processor defined in Claim 62 wherein the information specifies a location in a memory at which the range is stored (109).

With respect to claim 65, The rule engine content processor defined in Claim 47 wherein the search array comprises:

a plurality of rows of memory locations to store bytes of data (figure 1, P[j];

a plurality of rows of byte comparators (figure 1, c1-c4) to compare bytes of the data stored in the plurality of rows of memory locations with bytes of the pattern (e.g. "filters"), each comparator (figure 1, c1-c4) of the plurality of rows of byte comparators having an output (10, 12);

a plurality of masked reduction units, each of the plurality of masked reduction units coupled to receive byte masks and comparator outputs (e.g. a "HIT" or EROF") of comparators (c1-c4) in one row of byte comparators (e.g. P[1]), the plurality of masked reduction units masking individual comparator outputs based on the byte masks and combining unmasked comparator outputs for each row into one of a plurality of mask lines (figure 1, stage 1; e.g. Lee shows if an early out flag were encountered the remaining pattern (i.e. P[5-7] would not be search and in effect, masked).

With respect to claim 66, The rule engine content processor defined in Claim 47 further comprising:

a rule memory to store a plurality of rules (drawing reference 110);

a rule sequencer coupled to the rule memory to select one or more rules for execution (col. 7 line 49-53, figure 1); and

a decoder (110) to decode the one or more rules selected by the rule sequencer, the decoder coupled to the search array and sorter to provide decoded information to the search array and the sorter (col. 7 line 43).

With respect to claim 67, A process comprising:

loading a set of input payload search registers (figure 1, drawing reference 102, and s[j]) with content (col. 2 line 18-19);

storing a replication (col. 2 line 20-23 i.e. loading in parallel) of the content (col. 2 line 18-19) in the payload search registers (figure 1, drawing reference 102, and s[j]) in a search array (140) coupled to the registers (figure 1, drawing reference 102, and s[j]), wherein the search array (140) comprises:

a plurality of M rows figure 1, P[i]) of search array elements (figure 1, points of computation) coupled to a plurality of M output match lines (10, 12); and

a plurality of N columns of search array elements (figure 1, points of computation) coupled to a plurality of N pattern input lines (col. 5 line 57, col. 7 line 34-37) comprising a search pattern (col. 4 line 4; e.g. a target pattern), wherein the search array (140) comprises an array of M by N search array elements (figure 1, see the comparator array in each stage 1-3 as a two dimensional (i.e. M by N) array), and wherein the storing of the replication of content comprises (figure 1) replicating and storing the content in the M input payload search registers a plurality of times (col. 2 line 20-23 i.e. loading in parallel), in the plurality of N columns of search array elements (figure 1, points of computation);

presenting by means of the plurality of N pattern input lines a pattern identified by a search instruction to be searched in the search registers (figure 1; i.e. presenting "filters" to the input candidate stream);

performing parallel pattern matching (col. 2 line 26-27 and col. 4 line 15-17) between the pattern (e.g. "filters") and the content (col. 2 line 18-19) stored in the search array (140), wherein the parallel pattern matching comprises (col. 2 line 26-27 and col. 4 line 15-17) performing a simultaneous search (col. 7 line 1-6) query (col. 3 line 39) simultaneously within all M rows for the search pattern input by the N pattern input lines in one clock period (figure 1, stages 1, 2, or 3, col. 2 line 13-29, and col. 4 line 12-19; e.g. Lee discloses comparing all characters in "file" are compared with the target pattern in a single stage (i.e. a clock cycle); and

outputting by means of the plurality of M output match lines an indication of a result of performing the pattern matching (figure 1, e.g. a "HIT" indication or an EROF).

With respect to claim 68, The process defined in Claim 67 further comprising:

generating a plurality of match lines (figure 1, drawing references 10, 12, and col. 2 line 33-37, match trace; i.e. Lee discloses a match line for every point of computation) associated with rows (P[1-7]) of the search array (140), wherein match lines (figure 1, drawing references 10, 12, and col. 2 line 33-37, match trace; i.e. Lee discloses a match line for every point of computation) of the plurality of match lines indicating whether a match occurred between the pattern and data in a row associated with one of the match lines (i.e. an output of a "HIT" indicator);

performing the one or more operations (col. 6 line 1-4). on at least a set of the match lines (figure 1, drawing references 10, 12, and col. 2 line 33-37) in response to information specified by the search instruction (figure 2, instructions);

outputting an indication as to whether one or more of the match lines match the pattern and a result of performing the one or more operations (figure 1, "HIT" or "EARLY OUT").

With respect to claim 69, The process defined in Claim 67 wherein loading the search registers is performed to store, replicate, and interleave data such that data for one row is stored in an adjacent row in shifted form (figure 1, i.e. the match trace shows the shifting form of loaded data).

With respect to claim 70, The process defined in Claim 67 further comprising:

converting code into a sequence of search instructions (col. 7 line 43-44; i.e. decoding);

executing the sequences of search instructions in consecutive cycles such that pattern matching is performed for each of the plurality of search instructions (figure 2, instructions and figure 1, stages 1-3).

With respect to claim 71, A process for performing contextual searches in a pipelined fashion, the process comprising:

fetching a rule from a rule memory (figure 2, obtaining an instruction);

decoding the rule and assembling indirect fields (col. 7 line 43), if any;

storing a replication (col. 2 line 20-23 i.e. loading in parallel) of content (col. 2 line 18-19) in a plurality of input payload search registers (figure 1, drawing reference 102, and s[j]) in a search array (140) coupled to the registers (figure 1, drawing reference 102, and s[j]), wherein the search array (140) comprises:

a plurality of M rows figure 1, P[i]) of search array elements (figure 1, points of computation) coupled to a plurality of M output match lines (10, 12); and

a plurality of N columns of search array elements (figure 1, points of computation) coupled to a plurality of N pattern input lines (col. 5 line 57, col. 7 line 34-37) comprising a search pattern (e.g. "filters), wherein the search array (140) comprises an array of M by N search array elements (figure 1, see the comparator array in each stage 1-3 as a two dimensional (i.e. M by N) array), wherein the storing of the replication of content (figure 1) comprises replicating and storing the content in the M input payload search registers N times (col. 2 line 20-23 i.e. loading in parallel), once in each of the N columns of search array elements (figure 1, points of computation), wherein the row positions of the replicated content within each column of the N columns of search array elements is shifted relative to the row positions of the replicated content in each of its adjacent columns of search array elements (figure 1, i.e. the match trace shows the shifting form of loaded data);

executing one or more search operations (figure 2, instructions) on values in the search array (140); and

performing sort operations on results of executing the one or more search operations (figure 1, use of the 'V' register).

With respect to claim 72, The process defined in Claim 71 wherein the process is performed in a four stage pipeline (col. 1 line 55 and figures 1-2) with a search array (149) and a sorter (120).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 4-5, 7-8, 24, 33-34, 38, 49-50, and 61 are rejected as being unpatentable over Lee as applied to claims 1-3, 6, 9-23, 25, 27-32, 35-37, 39-48, 51-60, and 62-72 above in view of Messenger et al. (U.S. Patent 5,051,947).

With respect to claim 4 and similar claims 38 and 50, Lee does not expressly teach one parameter specifies a portion of the pattern to be masked to enable a subset of the pattern to be searched against the content in the search array.

Messenger, however, teaches one parameter specifies a portion of the pattern to be masked to enable a subset of the pattern to be searched against the content in the

search array (col. 10 lines 10-40; i.e. using a mask register) to mask out bits in a comparison with inputs.

In the same field of endeavor, (i.e. pattern matching), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the mask of Messenger would have given Lee the benefit of having a system that effectively masks out mismatched characters (as needed by Lee by use of an ("early out flag").

With respect to claim 5, Lee fails to expressly teach the portion of the pattern to be masked is specified by a mask vector to mask off specific bytes in the pattern.

Messenger, however, teaches the portion of the pattern to be masked is specified by a mask vector to mask off specific bytes in the pattern (col. 10 line 10-40 and col. 12 line 37) to mask out unwanted input characters.

With respect to claim 7 and similar claim 8, Lee fails to expressly teach instructions that specify a windowed-find-first-forward search and instructions that specify a windowed-find-first-reverse search.

Messenger, however, teaches instructions that specify a windowed-find-first-forward search and instructions that specify a windowed-find-first-reverse search (col. 24 line 15 and line 54) as a sliding window search for enabling the search functions to be performed only within a fixed window of one text segment.

In the same field of endeavor, (i.e. pattern matching), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the sliding window searching of Messenger would have given Lee the benefit of searching within boundaries of an input stream.

With respect to claim 24, Lee teaches the rule processor defined in Claim 21 wherein at least one search instruction in the one or more search instructions comprises opcode information to indicate a search operation type (col. 3 line 37-39; i.e. performing string searching or query types of searching), pattern information to specify a pattern to be located (e.g. "filters"), and a pair of offsets to specify starting and ending bounds of locations in the search registers for the search for the at least one search instruction (S[1-12]).

Lee fails to expressly teach a mask to specify a portion of the pattern information that comprises the pattern.

Messenger, however, teaches a mask to specify a portion of the pattern information that comprises the pattern (col. 10 lines 10-40; i.e. using a mask register) to mask out bits in a comparison with inputs.

In the same field of endeavor, (i.e. pattern matching), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the mask of Messenger would

have given Lee the benefit of having a system that effectively masks out mismatched characters (as needed by Lee by use of an ("early out flag").

With respect to claim 33, Lee teaches the rule processor defined in Claim 1 wherein the search array (140) comprises a first input to receive bits including the pattern (figure 1, e.g. "filters").

Lee does not expressly teach a second input for a mask, the search array to mask zero or more bits of the bits corresponding to the pattern based on the mask.

Messenger, however, teaches a second input for a mask, the search array to mask zero or more bits of the bits corresponding to the pattern based on the mask(col. 10 lines 10-40; i.e. using a mask register) to mask out bits in a comparison with inputs.

In the same field of endeavor, (i.e. pattern matching), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the mask of Messenger would have given Lee the benefit of having a system that effectively masks out mismatched characters (as needed by Lee by use of an "early out flag".

With respect to claim 34 and similar claims 49 and 61, Lee does not expressly teach wherein the bits including the pattern comprises N bytes and the mask comprises N bits, each of the N bits being associated with a different one of the N bytes, wherein the search array masks one of the N bytes of the pattern when its associated bit of the N bit mask is in a first state.

Messenger, however, teaches wherein the bits including the pattern comprises N bytes and the mask comprises N bits, each of the N bits being associated with a different one of the N bytes, wherein the search array masks one of the N bytes of the pattern when its associated bit of the N bit mask is in a first state (col. 10 lines 10-40; i.e. using a mask register) to mask out bits in a comparison with inputs.

In the same field of endeavor, (i.e. pattern matching), it would have been obvious to one of ordinary skill in the data processing art at the time of the present invention to combine the teachings of the cited references because the mask of Messenger would have given Lee the benefit of having a system that effectively masks out mismatched characters (as needed by Lee by use of an ("early out flag").

Response to Arguments

Applicant's arguments, see the remarks filed 10/11/2007, with respect to the rejection(s) of claim(s) 1, 67, and 71 under 102(b) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. Specifically, Applicant's arguments as to Messenger lacking disclosure a search array comprising an array of M by N search array elements (e.g. see page 21, last 8 lines of the remarks) in the present amendments have been persuasive. Furthermore, the Examiner agrees that Messenger does not substantially disclose the replicating and storing of the content as found in the present claims (e.g. see last paragraph of page 22 and top of page 23 of remarks).

For the at least the above reasons, the 102(b) rejection in view of Messenger has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Lee as presented in the rejection above.

The Examiner respectfully submits that Lee teaches the above features as presented in the claims. Further arguments are rendered moot in view of the new grounds of rejection.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert M. Timblin whose telephone number is 571-272-5627. The examiner can normally be reached on M-F 8:00-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Robert M. Timblin

Patent Examiner AU 2167

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100